VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NEREALISTICKÉ ZOBRAZENÍ VIDEA

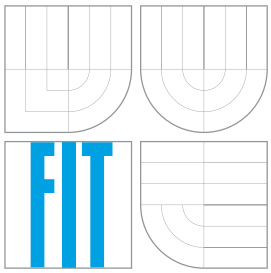
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL ŠIRŮČEK

BRNO 2012

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NEREALISTICKÉ ZOBRAZENÍ VIDEA

NON-REALISTIC VIDEO RENDERING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL ŠIRŮČEK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Dr. Ing. PAVEL ZEMČÍK

BRNO 2012

Abstrakt

Práce se zabývá způsoby pro nerealistické zobrazení videa. Jsou zde shrnuty techniky pro zpracování videa, hlavní zaměření je na malířské techniky a zpracování videa těmito technikami. Pozornost je věnována jednotlivým operacím při zpracování snímku a způsobům pro zajištění koherence mezi snímky videa. Součástí práce je aplikace na zpracování videa, využívající knihovnu OpenCV.

Abstract

The thesis deals a with ways of non-realistic view of video. There are summarized video processing techniques in this document. The main focus is in painting techniques and video processing with these techniques. Individual operations and image-processing methods, to ensure coherence between video frames, are also mentioned here. The application for video processing, using the OpenCV library, is the part of the thesis.

Klíčová slova

Nerealistické zobrazení videa, malířské techniky, zpracování videa, zpracování obrazu, gradient v obraze, OpenCV

Keywords

Non-realistic video rendering, painterly rendering, video processing, image processing, image gradient, OpenCV

Citace

Pavel Širůček: Nerealistické zobrazení videa, bakalářská práce, Brno, FIT VUT v Brně, 2012

Nerealistické zobrazení videa

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana doc. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny prameny a publikace, ze kterých jsem čerpal.

. . . . . .
Pavel Širůček
12. května 2012

Poděkování

Děkuji panu doc. Dr. Ing. Pavlu Zemčíkovi za odborné vedení a velmi rychlé zodpovídání položených dotazů.

Obsah

Seznam obrázků

Kapitola 1

Úvod

V počítačové grafice usilujeme nejčastěji o co nejpřesnější zobrazení reálného světa. Na druhé straně tohoto úsilí stojí oblast počítačové grafiky, která se zabývá zobrazením nereálným. Ať už se snažíme oprostit od detailů, které nás nezajímají a zdůraznit jiné skutečnosti nebo chceme vyvolat umělecký dojem, vždy jde o dobrovolný a záměrný odklon od realismu.

Tato práce se zabývá právě nerealistickým zobrazením se zaměřením na zpracování videosekvencí. Hlavním tématem budou techniky zaměřující se na vyvolání uměleckého dojmu a simulující práci člověka – malířské techniky. Součástí práce je aplikace, jejímž výstupem je nerealisticky zpracované video nebo obrázek.

Druhá kapitola vysvětluje, co to vlastně nerealistické zobrazení je, a jaký je důvod k jeho použití. Dále je vysvětleno zpracování videosekvencí cestou nerealismu a jeho největší problém – stabilita videa (či koherence mezi snímky).

Třetí kapitola je zaměřena na rozdělení technik pro nerealistické zpracování a zobrazení. Nejprve je uvedeno možné rozdělení metod nerealistického zpracování, poté jsou popsány různé druhy technik. Od simulace malby, přes metody mající výstup blížící se podobě animovaného filmu až po zmínku o simulaci kresby perem a technik spolupracujících s uživatelem.

Čtvrtá kapitola shrnuje způsoby zajištění koherence mezi snímky při zpracování videa malířskými technikami. Jsou uvedeny tři způsoby, jejich výhody a nevýhody a je navrženo řešení použité v této práci.

V páté kapitole je krok po kroku probráno zpracování vstupních snímků. Popsány jsou parametry využité při zpracování malby a jejich vliv na výsledný obraz nebo video.

Šestá kapitola popisuje použité nástroje a vlastní implemntaci. Je zde popsána využitá knihovna OpenCV, vlastní implementované třídy a popis výsledné aplikace.

V sedmé kapitole jsou uvedeny výsledky práce, shrnuty některé grafické výstupy a uvedena časová náročnost pro zpracování s různými parametry.

Kapitola 2

Nerealistické zobrazení

Nerealistické zobrazení je jednou z novějších oblastí počítačové grafiky a poměrně rychle se vyvíjí. V anglické literatuře se pro realistické zobrazení používá zkratka *PR* (*Photorealistic Rendering*) a pro nerealistické zobrazení *NPR* (*Non-Photorealistic Rendering*). Tyto zkratky budu používány i v této práci.

Metody PR usilují o zobrazení scény tak, aby byla nerozeznatelná od skutečného stavu. Existují ale situace, ve kterých nelze (foto)realismu dosáhnout, nevyžadujeme jej nebo je pro nás dokonce nežádoucí. Pokud je scéna příliš složitá a těžko se v ní vyznává, umožňuje nám NPR vynechat či zanedbat nepodstatné části a naopak podstatné a pro nás zajímavé části zvýraznit (barvy, obrysy, . . .). Pouze s pomocí např. šrafování můžeme zvýraznit zakřivení povrchů, které jinak není dostatečně patrné nebo přehledné.

Kromě údajů o vzhledu může scéna obsahovat také další dodatečné informace (teplota, rychlost pohybu, . . .). Tyto údaje lze pak zobrazit pomocí široké škály stylů, které NPR nabízí. Techniky NPR lze proto s úspěchem použít pro vizualizaci vědeckých dat, což je efektivní právě díky možnosti zvýraznění důležitých informací na úkor nedůležitých.

Při tvorbě animací lze NPR úspěšně kombinovat s ruční prací, kdy jsou složité scény (např. pozadí) vytvořeny na počítači a pak jsou zkombinovány s ručně kreslenými objekty (např. postavami). Styl prostředí vytvořeného v počítači se pak upravuje tak, aby výsledek po sloučení s ruční prací nepůsobil rušivě.

V určitých případech se NPR používá také z ekonomických důvodů, například v knihách, které jsou omezeny pouze na černobílé ilustrace. V tomto případě se často využívá technika tečkování, kdy obrázek vypadá lépe, než kdyby byl v odstínech šedi. [13]

Nerealistický styl je také v některých případech považován za atraktivnější než realistické zobrazení. Jsou to právě případy, kdy se snažíme simulovat práci umělce či kreslíře. NPR pak zvyšuje estetický dojem ze zpracovaného obrazu, přitahuje větší pozornost (využití v reklamě) či na první pohled ihned zaujme více než pouhá fotografie.

2.1 Zpracování videa

Nejjednodušším způsobem, jak zpracovat nefotorealisticky video, kdy můžeme použít téměř jakoukoliv techniku, je aplikace vybrané metody na každý snímek vstupního videa. Problém nastává v případě, kdy by byly snímky zpracovány nezávisle na sobě. Setkáváme se s pojmem stabilita metody (techniky) nebo s pojmem koherence snímků. Jelikož může být každý snímek vykreslen jinak, můžou se ve výstupním videu objevit nepříjemné ruchy či blikání. S tímto problémem se nejvíce setkáme u technik, kde se při vytváření nerealistického obrazu používá náhoda (napodobení práce

člověka). Náhodně umísťované tahy štětcem, hrbolaté obrysy či siluety, náhodně umístěné šrafování či tečkování, to vše je zdrojem problémů. Např. ve videu, kde je téměř statické pozadí, lišící se v následujících snímcích jen v detailech, můžou být na výstupu zaznamenány velké změny. Tyto změny jsou pro nás nežádoucí, neboť odvádí pozornost od hlavního obsahu videa. Při zpracování se tedy snažíme tyto ruchy alespoň minimalizovat, pokud bychom je chtěli úplně odstranit, nejspíše bychom museli upravovat ručně snímek po snímku. Ne každá technika pro NPR zpracování se tedy může hodit pro zpracování videa. Pro zajištění stability a udržení požadované výstupní kvality existuje mnoho způsobů, některé z nich budou popsány v dalších kapitolách.

Kapitola 3

Metody a techniky nerealistického zobrazení

V této kapitole bude uveden stručný přehled některých metod a technik pro nerealistické zpracování. Přehled je doplněn o obrázky ilustrující výstup u uvedených technik.

Metody NPR se dají rozdělit podle mnoha faktorů, zde je jeden ze způsobů [13]:

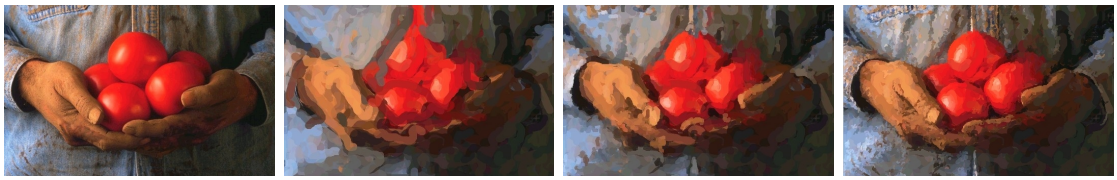
- podle míry interakce s uživatelem
  - plně interaktivní programy (uživatel např. určuje všechny tahy sám)
  - polointeraktivní programy (uživatel určuje kde se má kreslit nebo vytváří „šablonu“
  - plně automatické zpracování (např. použití specifických filtrů)
- podle druhu vstupních dat
  - 3D data (povrchová nebo objemová reprezentace)
  - 2,5D data (většinou 2D obrazy s přidanými informacemi o hloubce povrchových bodů)
  - 2D data (statické obrázky, snímky videa)
- podle toho zda pracujeme s bitmapou nebo vektory (některé NPR techniky mohou být schopny generovat vektorová data – použitelnost v jakémkoliv rozlišení)
- podle toho zda zpracováváme statický obraz nebo video (animaci)

Technik, jak přepracovat realistický obraz na nerealistický, existuje velké množství. Velký přehled pro NPR lze nalézt v [9]. Bohužel většina odkazů na této webové stránce už není aktuálních, ale jako přehled může posloužit velice dobře. Mnoho technik vychází ze stejného základu a liší se např. pouze v detailech nebo způsobu zpracování. Dále bude popsáno několik základních technik, spolu s ukázkami, jak může vypadat výstupní obraz nebo video.

3.1 Malířské techniky

Malířské techniky (v angl. painterly rendering) budou hlavním obsahem práce, proto jim bude věnováno více prostoru. Cílem při simulaci malířských technik je napodobení práce malíře a vyvolání dojmu ruční práce. Simulují se tahy štětcem. Tahy štětce jsou většinou reprezentovány rovnými čarami nebo křivkami. Směr tahu je obvykle určen gradientem v obrázku, zjištěným pomocí Sobelova operátoru. Zpracování obrázku je realizováno ve více průchodech. V prvním průchodu je

tloušťka štětce největší a je pokryto celé plátno. V dalších průchodech jsou použity tahy s menším průměrem a vykreslují se tam, kde je potřeba detailnější malba (např. v okolí hran). Počet průchodů pak může ovlivnit míru abstrakce obrázku, čím méně bude průchodů se zmenšováním tloušťky štětce, tím méně bude detailů ve výsledném obrázku. Obrázek 3.1 ukazuje způsob zpracování, kdy tahy štětce reprezentují křivky. 3.1a je originální obrázek, 3.1b je obrázek po prvním průchodu (tahy mají průměr 8), 3.1c je obrázek po druhém průchodu (tahy mají průměr 4) a konečně 3.1d je finální obrázek po třetím průchodu (tahy mají průměr 2).

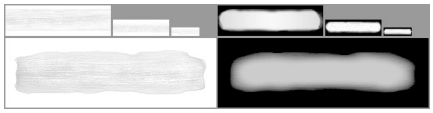


(a) (b) (c) (d)

Obrázek 3.1: Průběh zpracování obrázku podle [2].

Použitím různých parametrů pro tah štětce lze experimentovat a vytvářet obrázky v různých malířských stylech. Uvedený je styl *„Impressionist“* (normální styl malby, bez zvláštních parametrů), *„Expressionist“* (prodloužené tahy, upravené barvy) nebo styl *„Pointillist“* (tahy jsou reprezentovány kruhy s postupně se zmenšujícím průměrem).

Tahy štětce mohou být také reprezentovány texturou a maskou, která určuje jejich průhlednost. Tento způsob NPR je popsán v [1]. Na obrázku 3.2 jsou vidět textury tahů a jejich masky. V každém průchodu se opět používá tenší štětec. První vrstva tahů s největším průměrem je umístěna po celém obrázku. Tahy s menším průměrem jsou pak umísťovány na místa, kde jsou detekovány hrany nebo do jejich okolí. Směr tahu je opět určen gradientem, ale pouze v místech, kde je gradient vysoký. Na směr tahů je poté aplikována RBF a tím se zjistí směr tahů v ostatních místech. Výsledek tohoto způsobu malby je vidět na obrázku 3.3.



Obrázek 3.2: Textury reprezentující tvar tahu štětce. Převzato z [1].

Obrázek 3.3: Nanášení textur tahů štětce. Převzato z [1].

Další způsob zpracování s využitím textur je uveden v [7], viz. obrázek 3.4. Vstupy a výstupy

mohou být různé, na obrázku 3.4a je malba připomínající použití voskovek. Obrázek 3.4b ukazuje simulaci malby na snímku z videa. Obrázky 3.4c a 3.4d ukazují možnosti prezentovat 3D scénu pomocí malířských technik a kombinaci 3D objektu (ještěrka) s pozadím pastelkami či voskovkami.

(a) obrázek (b) snímek videa (c) 3D model (d) kombinace 3D a 2D

Obrázek 3.4: Různé výstupy malířských technik [7].

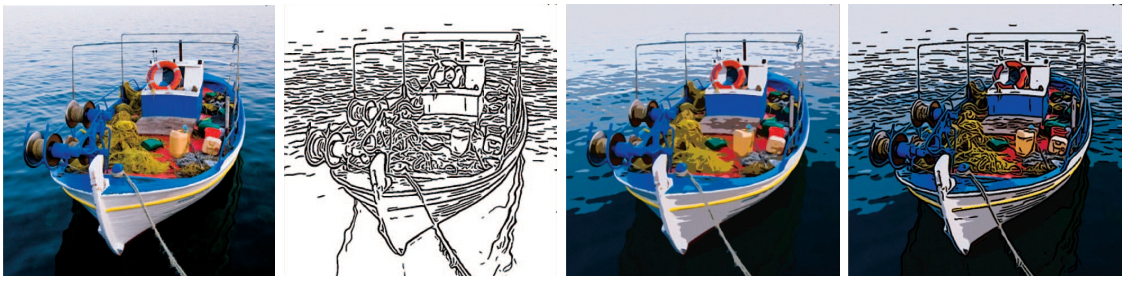
Zpracování videa pomocí malířských technik bude rozvedeno v další kapitole. Ve světě filmu byly tyto techniky použity například ve snímku Jak přicházejí sny (What Dreams May Come), pro různá pozadí a efekty (obrázek 3.5), ve kterých se pohybovali reální herci.

Obrázek 3.5: Obrázek z filmu Jak přicházejí sny (http://www.vfxhq.com/1998/dreams.html).

3.2 Cell shading, komiksové efekty

Tyto techniky jsou velmi často používané, ať už v počítačové grafice, počítačových hrách nebo filmech. Výsledek připomíná komiksovou malbu. V obrázku jsou upraveny barvy (je redukován jejich počet). Zvýrazněním a vyhlazením hran dosáhneme výsledného komiksového efektu, či efektu připomínajícího animovaný film (cartoon).

Jak může takové zpracování vypadat ukazuje obrázek 3.6, pocházející z [5]. Jako první krok je provedena abstrakce obrázku, mající za úkol jeho zjednodušení a zároveň zdůraznění nejvýznamnějších prvků. K tomuto úkonu je možné použít např. bilaterální filtr. Na abstrahovaném obrázku je dále provedena kvantizace (3.6c), sloužící k podtržení kresleného vzhledu. Jako další krok je provedena detekce hran (3.6b). K detekci je možno využít několik způsobů (jako základ např. rozdíl Gaussových rozostření). Výstup je dán zkombinováním abstrahovaného obrázku s detekovanými hranami (3.6d).



(a) (b) (c) (d)

Obrázek 3.6: Zpracování obrázku způsobem popsaným v [5].

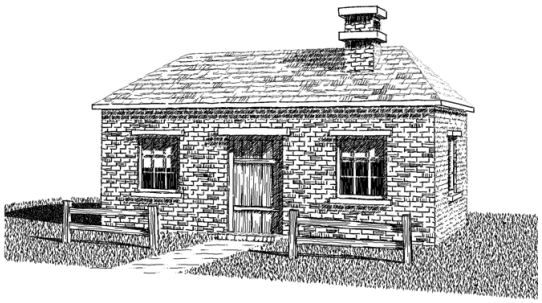
Pokud chceme těmito způsoby zpracovávat video, je zajištění koherence mezi snímky většinou jednodušší než např. při zpracování malířskými technikami. V [4] se vychází z toho, že snímky jsou zpracovávány vesměs nezávisle na sobě (předchozí snímky neovlivňují aktuální snímek). Je to proto, že snímky v nerealistickém podání se liší téměř stejně, jako snímky v realistickém podání, kdežto třeba u malířských technik nebo kresby perem se mohou projevit velké změny. Menší problém se může objevit při zvýraznění hran. Pokud jsou hrany ostré, mohou způsobovat blikání. Částečným řešením je reprezentace méně důležitých hran ve stupních šedi (a ne pouze v černé barvě). Pokud taková hrana zmizí, nepůsobí to tolik rušivě.

Těmito technikami byl v roce 2006 zpracován snímek Temný obraz (A Scanner Darkly). Film byl nejdříve natočen zcela normálním způsobem. Poté byl několik měsíců upravován na počítačích do animované podoby, upravovat se musely postupně jednotlivé snímky.

Obrázek 3.7: Obrázek z filmu Temný obraz (http://www.rottentomatoes.com/m/scanner_darkly/).

3.3 Simulace kreseb a rytin, perokresba

Narozdíl od simulace malby, kde získáme vyplněnou plochu, je výstupem u simulace kresby síť různých čar. Čáry mohou být uloženy ve vektorovém formátu. Nástroje simulované touto techikou jsou většinou ty, které zanechávají úzkou stopu (pero, tužka, rydlo, . . .). Kresba neobsahuje informace o barvě a jejím odstínu. Tyto vlastnosti (např. pokud chceme vybarvit plochu) musí být nahrazeny vhodným umístěním šrafování nebo tečkování. Kresba je zaměřena na hrany a jejich oddělení od dalších komponent obrázku, protože hrany nebo obrysy jsou v kresbě hlavním nositelem informace [13].



Obrázek 3.8: Ukázka perokresby uvedené v [11]

Ve zpracování videa není tato technika moc používaná. Spíše se využívá při zpracování samostatných obrázků nebo při animování 3D scény. V [12] a v [13] se autoři zabývají problémem při náhodném umísťování čar. Při animování vzniká rušivý šum těchto čar. Je tedy nutné zajistit časovou koherenci jejich umístění. Jednou z možností je držení čar na místě relativně k obrazu, to zaručí určitou koherenci, ale animace pak působí dojmem, že se odehrává za nepohyblivým reliéfním sklem (*shower-door* efekt). Lepší výsledky dává použití technik, které se snaží držet čáry relativně k objektům scény, které zobrazují. Je tak možné vytvořit například animaci rotující koule, na jejímž povrchu se čáry pohybují, jako by tam byly nakreslené.

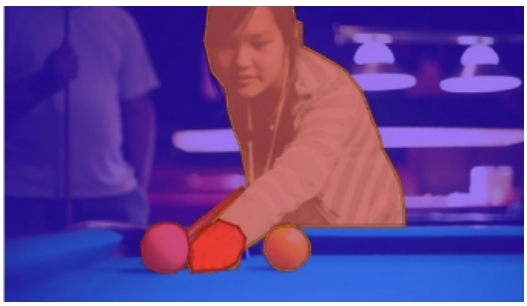
Žádné řešení však není ideální. Ruchy nebo artefakty se buď pohybují příliš chaoticky nebo naopak příliš uspořádaně a narušují tak vnímání vlastního objektu.

3.4 Interaktivní techniky

Pokud může uživatel ovlivnit grafický výstup, hovoříme o interaktivních technikách. Uživatel může například zvolit v obrázku oblast, kterou chce nerealisticky vyrenderovat, může zvolit postavy, které chce nějakým způsobem zvýraznit, objekty na které se chce zaměřit atd.

O interaktivním způsobu zpracování videa pojednává článek [10]. Uživatel označí ve vybraných snímcích klíčové regiony. Tyto regiony jsou pomocí metody mean-shift interpolovány i v ostatních snímcích. Regiony jsou následně extrahovány a uživatel si může zvolit např. jiné pozadí.

V [8] byl představen systém AniPaint. Jde o aplikaci pro zpracování videa malířskými technikami. Uživatel ve snímcích sám označuje regiony, které chce animovat (např. postavy). Regiony, které nejsou označeny (např. pozadí, které se může měnit) nemění svou podobu. Uživatel může takto označit např. pohybující se postavu, viz. obrázek 3.9a. Dále může uživatel sám určovat tahy štětců, viz. obrázek 3.9b. Výstupy z tohoto programu pak vypadají velmi profesionálně a na první pohled zaujmou.



(a) Vyznačené regiony

(b) Uživatelem definované tahy štětce

(c) Snímek z videa (socha ve filmu Pán prstenů) zpracovaného v programu

Obrázek 3.9: Zpracování animací v programu AniPaint [8].

Kapitola 4

Techniky zpracování videa

V předchozí kapitole byly uvedeny některé způsoby zpracování obrazu nerealistickým způsobem. V této kapitole budou tyto způsoby rozšířeny o zpracování videa při použití malířských technik.

Jak již bylo uvedeno, největší problémy způsobuje při zpracování videosekvencí náhoda. V případě simulace malby, kde je hlavním úkolem napodobit práci člověka, je tento problém o to větší.

4.1 *Optical flow*

Zpracování snímků za využití optického toku. Nejspíše první známější ze způsobů, jak zajistit koherenci mezi snímky je popsán v [6].

- Jako tahy štětce jsou použity přímky, směr přímky je určen gradientem.
- Tahy jsou umísťovány v pravidelné mřížce (mezi tahy jsou konstantní mezery).
- Při malbě se využívá detekce hran. Tah štětce se ukončuje právě v místě, kde je detekována hrana.
- Barva tahu je dána barvou bodu v původním obrázku (na souřadnicích jako je počátek tahu).
- Při zpracování videa je plně namalován pouze první snímek. V každém dalším snímku je pak počítán optický tok mezi aktuálním a předchozím snímkem. Výstupem je pole vektorů optického toku, podle kterého jsou tahy posunuty na nové místo. Při posouvání dochází k situaci, kdy některé části snímku nejsou plně pokryty tahy štětce a v některých částech se tahy zbytečně hromadí. V prázdných místech jsou potom vykresleny nové tahy za použití Delaunayho triangulace (využívá středy tahů umístěných v předchozím snímku před posunem pomocí optického toku). V místech, kde je tahů zbytečně moc, se některé tahy odstraní.
- Tato metoda selhává v případě, kdy dojde k velké změně ve vstupním videu.
- Autor udává i průměrnou dobu zpracování jednoho snímku – 81 sekund na PC běžícím na frekvenci 180 MHz (rok 1997).

Obrázek 4.1: Dva po sobě jdoucí snímky a optický tok mapující pohyb pixelů mezi nimi [6].

Možného rozšíření se tento způsob dočkal v [7], kde je gradient počítán ze tří snímků (z aktuálního a dvou předchozích). Výsledný gradient je dám zprůměrováním gradientů z každého ze snímků. K posunu tahů je pak využit právě optický tok.

4.2 *Paint-over*

Tento způsob byl uveden v [3] a jde o jednoduchou cestu, jak zajistit základní koherenci mezi snímky videa.

- Jako tahy štětce jsou použity křivky, dají se však použít i přímky.
- Tahy nejsou narozdíl od předchozího způsobu umísťovány do mřížky, ale tam, kde se aktuální obraz liší od obrazu zpracovaném v předchozím průchodu. Do částí s minimálními změnami se nezasahuje a jsou ponechány bez úpravy.
- Při zpracování videa je opět první snímek namalován kompletně. Rozdíl nastává při zpracování dalších snímků. Každý následující snímek totiž vychází z předchozího a na ten se nanáší nové tahy, právě v místech změny oproti předchozímu snímku. Vždy je tedy nutné mít předcházející namalovaný snímek uložený v paměti.
- Tato metoda dává poměrně dobré výsledky, v případě menších změn ve videu. Pokud dojde rychle k velké změně mezi snímky, tato metoda, stejně jako předchozí, selhává. Naopak v případě videa se statickým pozadím (kamera je pevně umístěna na místě) a pohybujícími se objekty, přináší slušné a zajímavé výsledky.

4.3 *Difference masking*

Další způsob je uveden stejně jako předchozí v [3].

- Technika spočívá v tom, že se snímek deformuje (warping) v následující, a to samé se provede s body (v tomto případě kontrolními body křivek reprezentujících tahy), kterými vede tah štětce. Uvedený algoritmus zpracování vypadá následovně:

Paint the first frame
for each successive frame:
    Warp previous source frame to current for difference mask
    Warp all brush stroke control points to the current frame
    Paint the current frame over the warped painting

- V případě tohoto způsobu je nutné v paměti uchovávat i jednotlivé tahy v předchozím snímku. Paměťově jde tedy o náročnější metodu.

Obrázek 4.2: Zpracování videa pomocí technik optical flow, paint-over a difference masking. Převzato z [3].

4.4 Shrnutí

Základem většiny technik je využití optického toku mezi snímky. Tato technika dává samostatně vcelku dobré výsledky. Jak již bylo řečeno, pokud se tahy pouze posunují pomocí optického toku, vznikají prázdná místa, kde se musí generovat nové tahy. Technika paint-over je nejjednodušší na implementaci, ale samostatně dává asi nejhorší výsledky. Použití difference masking je zase paměťově nejnáročnější, což ale v dnešní době nepředstavuje výraznější problém. Pro dosažení co nejlepších výsledků, je pak vhodné tyto techniky kombinovat.

4.5 Návrh řešení

Ve své práci jsem se nakonec rozhodl o kombinaci postupů při zpracování malby, a zajištění koherence technikou paint-over. Při zpracování se bude uchovávat vždy předchozí snímek. Ten bude použit jako základ při malbě aktuálního a všechny tahy se budou umísťovat do něj. Jako tahy štětce bude možné zvolit přímky, křivky nebo tahy složené z přímek. Začátky tahů budou umísťovány v pravidelné mřížce (a podle odlišnosti od předcházejícího průchodu nebo snímku), dané parametrem. Vzdálenost mezi tahy se bude každým průchodem snižovat. Stejně tak délka nebo průměr tahu bude s každým průchodem nižší.

Kapitola 5

Postup zpracování vstupního videa

V této kapitole bude popsán postup vytváření nerealistického videa a operace aplikované na vstupní snímky. Budou nás zajímat následující věci:

- Rozostření snímku (abstrakce, odstranění šumu)
- Výpočet gradientů v požadovaných místech, zjištění orientace tahu štětce a barvy tahu
- Vykreslení tahů štětce na plátno
- Zajištění koherence mezi snímky videa
- Detekce hran ve snímku
- Vliv parametrů na zpracování malby

5.1 Rozostření snímku

První částí při zpracování snímku je jeho rozostření pomocí Gaussova filtru. Rozostření pomocí tohoto filtru při malbě bylo uvedeno v [2]. Toto rozostření abstrahuje obrázek (odstranění detailů) a redukuje šum ve vstupním snímku. Čím silnější filtr použijeme, tím více ztratíme detailů. U simulace malířských technik má abstrakce menší význam než u jiných technik, neboť nanášení tahů štětce snímek abstrahuje v nezanedbatelné míře samo o sobě.

Gaussův filtr je pravděpodobně nejpoužívanější rozostřovací filtr (ale není nejrychlejší). Filtr konvoluje každý pixel vstupního obrazu s gaussovým jádrem (konvoluční maska) a jejich suma dává výsledný obraz.

Ve dvou rozměrech má rovnice gaussovy funkce následující tvar:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

kde x je horizontální vzdálenost od počátku, y je vertikální vzdálenost od počátku a σ je směrodatná odchylka gaussovy distribuční funkce. Hodnoty z této rovnice jsou použity pro vytvoření konvoluční masky. Pro každý pixel obrázku je potom vypočítána nová hodnota zprůměrováním hodnot sousedních pixelů.

Při zpracování snímků je gaussovo rozostření použito několikrát. Poprvé se použije před prvním průchodem s maskou velikosti 15×15. Dále se snímek s nanesenými tahy rozostřuje po každém průchodu, ovšem už s menší maskou (3×3).

(a) originál

(b) rozostření gaussovým filtrem 15×15

Obrázek 5.1: Gaussovo rozostření

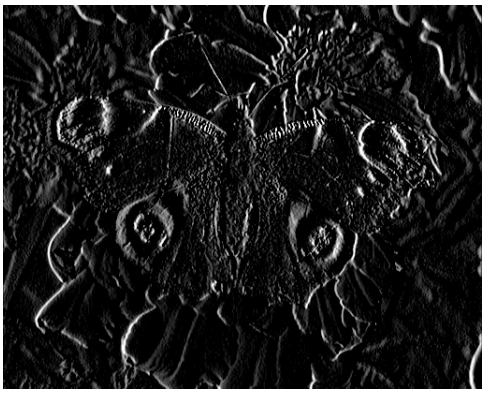
5.2 Výpočet orientace tahu štětce a barvy

Jak již bylo uvedeno, orientace tahu je určena, při použití malíšků technik, gradientem v obrázku [2]. Gradient je určen pomocí Sobelova operátoru. Základní tvar Sobelova operátoru je reprezentován dvěma maticemi (konvolučními maskami), a to ve vodorovném a svislém směru. S oběma maticemi je provedena konvoluce se vstupním obrázkem ($\mathbf{A}$):

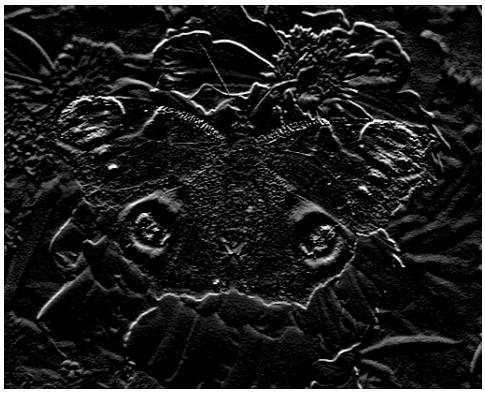
$$\mathbf{G_x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * \mathbf{A} \qquad \mathbf{G_y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * \mathbf{A}$$

Velikost gradientu v místě $[x, y]$ je pak určena následovně:

$$\mathbf{G} = \sqrt{\mathbf{G_x}^2 + \mathbf{G_y}^2}$$



(a) Aplikace ve vodorovném směru



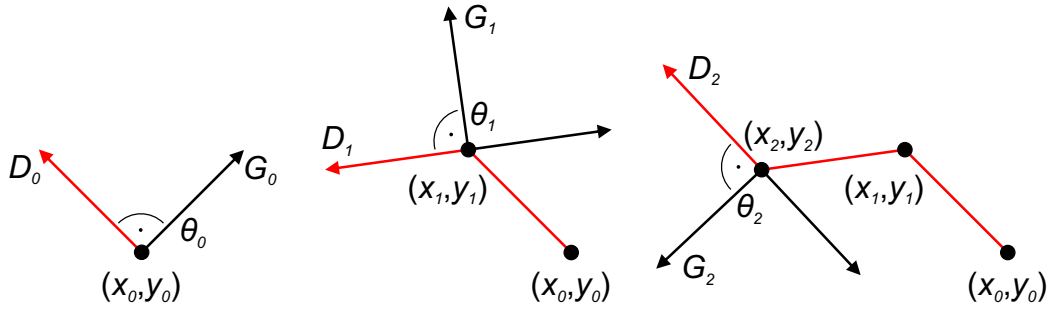
(b) Aplikace ve svislém směru

Obrázek 5.2: Aplikace Sobelova operátoru na obrázek.

Směr výsledného tahu je pak kolmice ke zjištěnému gradientu. Kolmici ke gradientu získáme jednoduše, a to prohozením souřadných os a změnou znaménka jedné osy. Pokud tahy štětce

reprezentují přímky, nic nám nebrání tah vykreslit. V případě použití křivek nebo skládání přímých tahů je nutno počítat dále. Ke gradientu existují vždy dvě kolmice, zvolíme tu, která způsobí menší zakřivení, viz obrázek 5.3. $[x_0, y_0]$ jsou souřadnice zkoumaného pixelu (počátek tahu), G je gradient, θ je směr gradientu zjištěný pomocí Sobelova operátoru a D je kolmice ke gradientu.

Vzdálenost mezi souřadnicemi kontrolních bodů (v případě použití přímek délka celého tahu) je určena vstupním parametrem. Počet kontrolních bodů (délka složeného nebo zakřiveného tahu) je pak dán dalšími parametry (minimální a maximální počet těchto segmentů).



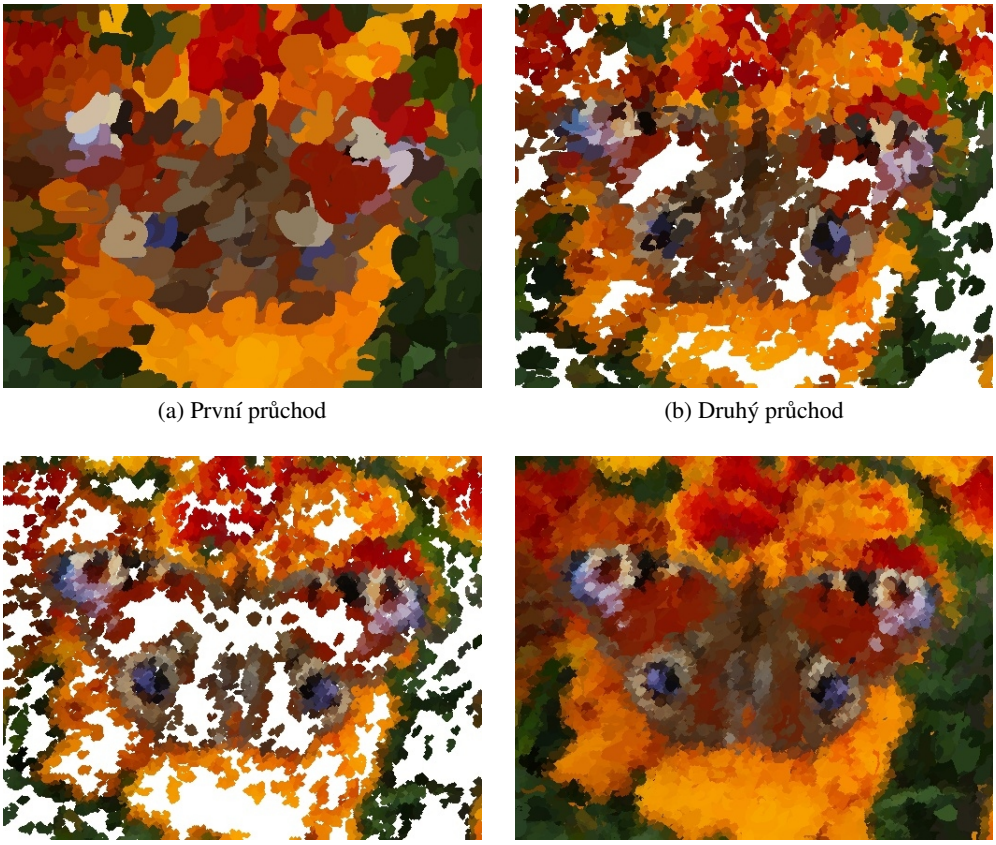
(a) (b) Zvolený směr D1 (c) Zvolený směr D2

Obrázek 5.3: Počítání směru tahu a kontrolních bodů tahu štětce. Inspirováno obrázkem v [3].

Barva vykresleného tahu je, v případě jednoduchých tahů, dána průměrnou hodnotou barev v počátečním a koncovém bodu tahu. U ostatních druhů tahů je barva dána pouze barvou počátečního bodu, protože v případě dlouhých tahů by se průměrná hodnota mohla velmi lišit od původní barvy, a snímek by tím utrpěl na kvalitě. Délka složeného nebo zakřiveného tahu je tedy omezena nejen parametrem, ale také prudkou změnou barvy v dalším kontrolním bodu (pokud se liší o víc než zadaný práh, ale zároveň se dosáhlo minimální požadované délky, tah končí).

5.3 Vykreslení tahů štětce na plátno

Pokud jsou všechny tahy vypočítány, zbývá je jen vykreslit na plátno. Tahy jsou vykreslovány v několika vrstvách podle zadaného počtu průchodů (obvykle stačí 3 průchody). V první vrstvě jsou vykresleny tahy s největším průměrem a je pokryto celé plátno (v případě zpracování prvního snímku). V dalších průchodech se postupně snižuje průměr štětce a tahy už nejsou vykreslovány na celé plátno, ale pouze tam, kde je potřeba většího detailu – hlavně v místech obrysů nebo v místech, kde je barva tahu z předchozího průchodu odlišná od barvy rozostřeného obrázku (míra odlišnosti, kdy ještě není potřeba tah vykreslovat, nebo už je nutné ho vykreslit, je určena parametrem *práh (threshold)*). Jednotlivé průchody jsou ukázány na obrázku 5.4. Délka tahu použitá v obrázcích je 24px a po každém průchodu se snižuje na polovinu. Tloušťka štětce je 12px a po každém průchodu je snížena na polovinu. Krok průchodu obrázkem je 12px a v každém průchodu je snížen o 4px.



(a) První průchod

(b) Druhý průchod

(c) Třetí průchod, umísťování posledních nejdrobnějších tahů

(d) Konečný obrázek

Obrázek 5.4: Zpracování snímku či obrázku třemi průchody.

5.4 Koherence mezi snímky

Výsledná kvalita videa je při nerealistickém zpracování nejvíce posuzována podle toho, jak na sebe jednotlivé snímky navazují. Tento problém jsem se pokusil řešit metodou překreslování snímku pouze v místech, kde se liší od předchozího. Při zpracování je kompletně namalován pouze první snímek a každý další vychází z přechozího. Je tedy nutné uchovávat vždy poslední snímek v paměti a porovnávat jej s aktuálně zpracovávaným snímkem. Nevýhodou tohoto způsobu je, že i nevýrazné změny, které jsou téměř nepostřehnutelné, stejně způsobují generování nových tahů. Záleží pak na obsahu videa.

5.5 Detekce hran

Detekce hran je v našem případě volitelnou částí. U malířských technik není moc důvodů pro zvýrazňování hran. Přesto, pokud jako poslední krok při renderování snímku zvýrazníme hrany, dosáhneme zajímavého efektu, kdy přes přesné hrany přesahují všemi směry tahy štětce.

Detekce hran je prováděna pomocí rozdílu dvou Gaussových rozostření [4]. Na originální obrázek je aplikován Gaussův filtr s velkým jádrem, poté je opět na originální obrázek aplikován filtr s menším jádrem. Hrany pak dostaneme odečtením druhého obrázku od prvního. V implementaci

jsou použity jádra velikosti 21×21 a 3×3.

(a) Rozdíl gaussových rozostření. Obrázek je po inverzi barev.

(b) Výsledná malba se zvýrazněním hran.

Obrázek 5.5: Zvýraznění hran ve výsledném obrázku.

5.6 Parametry malby

Vytváření nerealistického videa nebo obrazu je ovlivněno několika parametry, které zde budou popsány. Parametry jsou nastavitelné.

- *Tvar tahu štětce* – Tah štětce je možné reprezentovat třemi způsoby, přímkou (označení basic), několika přímkami poskládanými za sebe (long) nebo křivkou (curved). Jak mohou jednotlivé tahy vypadat ukazuje obrázek 5.6.

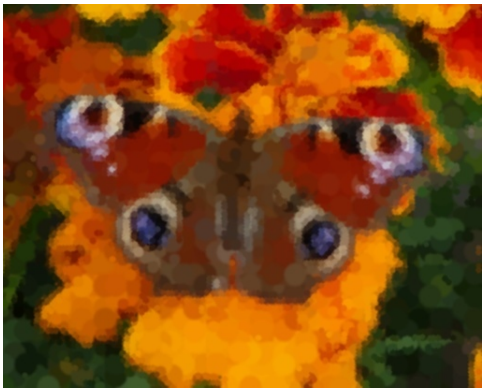
(a) Tah jako přímka (basic)

(b) „Složená“ přímka (long)

(c) Křivka (curved)

Obrázek 5.6: Druhy tahu a jejich možná podoba.

- *Délka segmentu tahu* – Pokud je jako tah použita jednoduchá přímka, označuje tento parametr jednoduše délku tahu. Pokud jsou použity jiné tahy, označuje parametr délku jednotlivých segmentů (vzdálenost mezi jednotlivými kontrolními body). Délka tahu je po každém průchodu snížena o zadanou hodnotu. Změnou délky je možné dosáhnout různých zajímavých výstupů, jak ukazuje obrázek 5.7. Příliš velká hodnota délky (navíc pokud není v každém průchodu snižována) se hodí spíše ke zpracování samostatných obrázků. Ve videu působí velmi dlouhé tahy ve snímcích rušivě. Doba zpracování jednoho snímku je přímo úměrná hodnotě délky (u složených tahů a křivek).



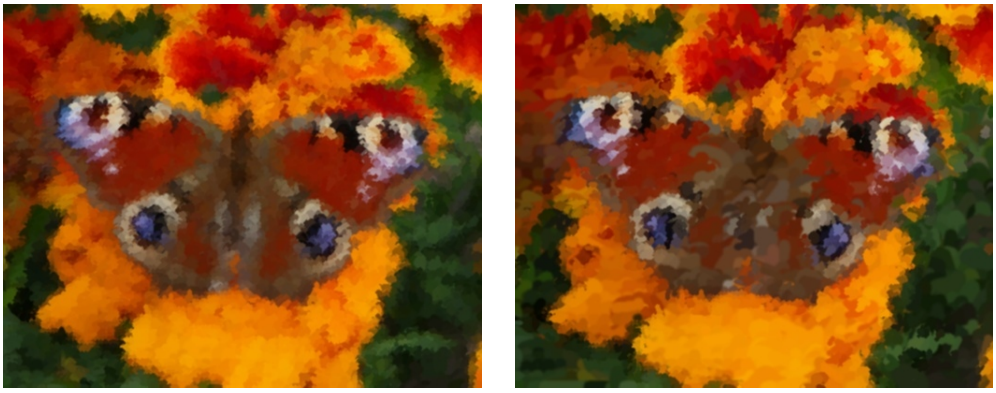
(a) Délka tahu je všech průchodech 1. Průměr je 16,8,4. Tahy jsou barevné kruhy.

(b) Délka tahu je ve všech průchodech 24. Průměr je 4,2,1.

Obrázek 5.7: Snímky s různou délkou tahu.

- *Maximální délka tahu* – Tento parametr je využit pouze u složených tahů a křivek. Udává maximální počet segmentů tahu.
- *Minimální délka tahu* – Stejně jako předchozí parametr, pouze udává minimální počet segmentů tahu. Tyto dva parametry mají také vliv na dobu zpracování jednoho snímku. Příliš vysoká hodnota minimálního počtu segmentů vede k dlouhému zpracování.
- *Tloušťka štětce* – Průměr vykresleného štětce. Stejně jako délka je po každém průchodu snížen o zadanou hodnotu.
- *Krok průchodu snímkem* – Parametr udává velikost kroku při průchodu maticí obrázku.
- *Počet průchodů* – Parametr udává, kolikrát je snímek zpracován, viz. obrázek 5.4. Každým průchodem se snižují parametry *délka segmentu*, *tloušťka* a *krok*, a tím se zjemňuje malba. Čím více průchodů, tím více se obrázek přibližuje originálu.
- *Práh* – Práh (či např. míra detailu) je jedním z nejdůležitějších parametrů při simulaci malby. Prahem je uvažována hodnota určující, zda bude vypočítaný tah vykreslen nebo ne. Je to maximální možný rozdíl barev originálního rozostřeného snímku a snímku po jednotlivých průchodech. Pokud se barva tahu namalovaného obrázku liší od barvy pixelu na témže místě v originálním obrázku o hodnotu nepřesahující práh, tah je zahozen a nevykresluje se. Pokud je rozdíl větší než hodnota prahu, tak se tah vykreslí. Nízkou hodnotou prahu se snímek přibližuje originálnímu snímku.[1] Vysoká hodnota prahu zase činí obrázek abstraktnějším, kdy se ztrácí detaily původního snímku. Velikosti prahu je úměrná doba zpracování snímku, čím je práh menší, tím více tahů je generováno (neplatí však vždy).

[1] V tomto případě ovšem záleží i na ostatních parametrech, jako je např. počet průchodů nebo počáteční tloušťka štětce.



(a) Hodnota prahu = 10 (b) Hodnota prahu = 50

Obrázek 5.8: Výsledky malby s různým prahem.

- *Pořadí tahů při vykreslení* – Parametr udává, v jakém pořadí budou tahy vykresleny na plátno. Jelikož jsou tahy štětce umisťovány v mřížce, a jsou počítány od levého horního rohu obrázku, do pravého dolního rohu, nastává situace, kdy je většina tahu překryta následujícím tahem. Tento problém je nevíce patrný u použití přímek. Malba pak nemusí působit příliš dobrým dojmem, což je nejvíce patrné u počátečních průchodů, další průchody pak tuto situaci víceméně řeší. Problém však může přetrvat ve větších plochách, kde už nejsou jemnější tahy štětce nanášeny. Tuto situaci jsem se pokusil řešit změnou pořadí vykreslování tahů. Tahy je možné seřadit podle jasu, a to buď od nejsvětlejších po nejtmavší nebo naopak. Výhodou těchto postupů je stále stejný způsob řazení i v následujících snímcích, takže se dále nezhoršuje stabilita mezi snímky. Rozdíly však nejsou zase tolik velké a jsou vidět spíše ve statických obrázcích, ve videu si pořadí vykreslení pozorovatel většinou nevšimne. Posledním způsobem je seřazení tahů náhodně, nicméně tento způsob je vhodnější spíše pro zpracování samostatných obrázků, u videa způsobuje další ruchy v obraze.

- *Zvýraznění hran* – Parametr udává, zda budou ve výsledném snímku zvýrazněny hrany, viz obrázek 5.5. Hrany jsou zvýrazněny po posledním průchodu. Zvýraznění hran se nehodí do každého videa nebo obrázku, v některých případech však zvýrazněné hrany přináší zajímavé výsledky. Mimo jiné tyto hrany kryjí rozdíly mezi snímky, a obraz působí méně rušivě, protože největší změny mezi snímky působí právě pohyb hran (pohyb objektů, postav, . . .) Zvýrazněním hran se dá tedy dosáhnout větší stability videa.

Další parametry

- *FPS* – Počet snímků za sekundu. Tento parametr se samozřejmě týká jen zpracování videa. Zatímco realistická videa mají okolo 30 snímků za sekundu, aby nebyla narušena plynulost, u nerealistického zpracování může být tato hodnota menší. V [3] je udávána doporučená hodnota 12 až 15 snímků za sekundu.

Kapitola 6

Implementace aplikace

V této kapitole budou popsány použité nástroje při tvorbě aplikace, vlastní implementace a funkce aplikace. Nakonec bude uveden formát parametrů pro zpracování malby.

Aplikace je konzolová, bez grafického uživatelského rozhraní. Pokud by bylo GUI přítomno, jednalo by se o jediné okno aplikace, umožňující načíst vstupní soubor, nastavit parametry a spustit zpracování. Nastavení parametrů je prováděno pomocí zadaného konfiguračního souboru.

6.1 Knihovna OpenCV

OpenCV (Open Source Computer Vision) je multiplatformní knihovna funkcí určená pro počítačové vidění, původně vyvíjená společností Intel. Obsahuje velké množství algoritmů pro manipulaci s obrazovými daty a pro zpracování obrazového vstupu v reálném čase. Veškeré zpracování obrázků nebo snímků videa v této práci je prováděno právě funkcemi z této knihovny. V implementaci je využito API psané v jazyce C, díky kterému by měl být program kompatibilní i se staršími verzemi knihovny. Implementace využívá verzi 2.3.1. Menší nevýhodou je to, že knihovna neumožňuje pracovat se zvukem. Výstupní video je tedy bez zvukové stopy, což je ale v případě zpracování videa s nižším počtem fps bezpředmětné.

Dále budou stručně popsány funkce a struktury knihovny použité v této práci:

Práce s obrázky

Strukturou pro uložení obrázku je `IplImage`. Tato struktura obsahuje všechny informace o obrázku (velikost, barevnou hloubku, . . .). Obsahuje ukazatel `*imageData` přes který je možné se dostat k barevným složkám obrazového bodu. Barevné složky pixelu v obrázku jsou uloženy v pořadí BGR, proto se na tuto skutečnost musí dávat pozor při přístupu k barvě. Obrázek je načten ze souboru pomocí funkce `openImage()`, uložen do souboru je pomocí funkce `cvSaveImage()`, případně může být v programu vytvořen pomocí funkce `cvCreateImage()` nebo zkopírován funkcí `cvCloneImage()`. Pro uvolnění obrázku z paměti je tu funkce `cvReleaseImage()`. Následující kód ukazuje použitý přístup k jednotlivým barevným složkám pixelu v obrázku na souřadnicích `[row, col]`.

```
IplImage *img = openImage("image.jpg");
uchar *data = (uchar *) img->imageData;
int step = img->widthStep / sizeof(char);
int channels = img->nChannels;
CvScalar pixelColor;
pixelColor.val[0] = data[row * step + col * n + 0]; /*modra*/
pixelColor.val[1] = data[row * step + col * n + 1]; /*zelena*/
pixelColor.val[2] = data[row * step + col * n + 2]; /*cervena*/
```

Práce s videosoubory

Při zpracování videa se využívá struktura `CvCapture` které je předán soubor s videem pomocí funkce `cvCreateFileCapture()`. Jednotlivé snímky videa jsou pak získávány funkcí `cvQueryFrame()`. Uvolnění souboru má nakonec na starost funkce `cvReleaseCapture()`. Pro uložení videa je použita struktura `CvVideoWriter` vytvořená voláním příslušné funkce `cvCreateVideoWriter()`, která má mimo jiné jako jeden z parametrů kodek výstupního videa. Jednotlivé snímky jsou do výstupního souboru zapisovány funkcí `cvWriteFrame()`. Uvolnění struktury je prováděno funkcí `cvReleaseVideoWriter()`.

Funkce pro zpracování obrazu

Pro konverzi barev je použita funkce `cvCvtColor()`. Sobelův filtr je implementován funkcí `cvSobel()`, rozostření funkcí `cvSmooth()`. Při detekci hran je využita funkce pro rozdíl snímků (matic) `cvSub()` a pro inverzi barev `cvNot()`.

6.2 Vlastní implementace

V této sekci budou popsány implementované třídy, jejich popis a nejdůležitější metody.

Funkce `main()`

Vstupní bod programu. Ve funkci main je provedeno zpracování parametrů příkazové řádky. Program vyžaduje při spuštění 3 parametry: vstupní soubor s obrázkem či videem, název výstupního souboru a soubor s parametry (bude popsán dále). V této funkci jsou také načteny parametry ze souboru a otevřeny či zavřeny vstupní a výstupní soubory. Ve smyčce jsou načítány pomocí třídy `VideoReader` jednotlivé snímky videa, které zpracovává třída `Paint` a poté jsou ukládány do výstupního souboru.

Třída `VideoReader`

Pomocná třída pro načtení videosouboru a uložení jeho parametrů. Obsahuje metody pro načtení videa, načtení snímku z videa (`getNextFrame()`) či pro zjištění parametrů videa.

Třída `Params`

Třída, ve které jsou uloženy parametry pro zpracování malby. Parametry jsou načteny ze souboru. Obsaženy jsou metody pro zjištění jednotlivých parametrů, úpravu parametrů po každém průchodu či navrácení parametrů do původního stavu před zpracováním snímku.

Třída `Paint`

Třída provádějící zpracování snímku. Vytváří několik obrázků, které se používají při zpracování (šedotónový, rozostřený a filtrovaný Sobelovým filtrem). Metoda prochází snímek zadaným krokem a vypočítává tahy štětce, které ukládá třída `Canvas`. Po každém průchodu jsou tahy vykresleny na plátno. Třída se také stará o zvýraznění hran po posledním průchodu snímkem, pokud je zvýraznění vyžadováno. Metoda `paint()`, která všechny tyto operace provádí, pak vrací namalovaný snímek (ukazatel na `IplImage`).

Třída `Canvas`

Třída implementující malířské plátno. Obsahuje seznam všech tahů aktuálního kroku, které je možné metodou `paintCanvas()` vykreslit do obrázku. Dále obsahuje seznam pořadových čísel jednotlivých tahů, který určuje pořadí, v jakém budou tahy vykresleny. K promíchání pořadí (aby bylo možné tahy vykreslovat i jinak, než z levého horního rohu do dolního pravého) slouží metody pro řazení podle jasu barvy štětce (`sortToLight()` a `sortToDark()`) nebo metoda pro náhodné uspořádání tahů (`randomize()`).

Třída `Stroke`

Třída pro tah štětce. Obsahuje seznam bodů, kterými je tah veden a barvu štětce. Vykreslení tahu je prováděno metodou `drawStroke()`, v případě použití křivek metodou `drawCurvedStroke()`.

Třída `Bspline`

Třída pro výpočet jednotlivých bodů křivky. Použita v případě, kdy chce uživatel tahy štětce reprezentovat křivkami. Metoda `compute()` bere jako parametry kontrolní body a počítá jednotlivé pixely křivky.

6.3 Spuštění aplikace

Aplikace se spouští přes příkazovou řádku a vyžaduje tři parametry uvedené krátkým přepínačem. Za přepínačem `-i` následuje vstupní soubor s videem (formát .avi) nebo obrázkem. Za přepínačem `-o` následuje název výstupního souboru (není nutné zadávat příponu). Poslední parametr `-p` uvozuje textový soubor s parametry pro zpracování. Příklad spuštění souboru:

```
videopaint.exe -i vstup.avi -o vystup -p soubor_parametru
```

Parametry zpracování

Soubor s parametry (podrobný popis v sekci 5.6) pro zpracování videa nebo obrázku vyžaduje následující formát:

```
type=CURVED
sort=TO_LIGHT
length=20
length_dec=0
thickness=12
thickness_dec=0
step=12
step_dec=4
pass=3
threshold=20
minL=2
maxL=18
edges=false
fps=15
```

- `type` udává typ tahu štětce, možné hodnoty jsou `BASIC`, `LONG` a `CURVED`.
- `sort` udává seřazení tahů před vykreslením, možné hodnoty jsou `NONE`, `TO_LIGHT`, `TO_DARK` a `RANDOM`.
- `length` udává délku tahu (segmentu) štětce.
- `length_dec` udává hodnotu, o kterou bude délka segmentu snížena po každém průchodu. Pokud je hodnota 0, délka tahu se po průchodu nesnižuje, záporná nebo hodnota větší než délka, snižuje délku po každém průchodu na polovinu.
- `thickness` udává průměr tahu štětce.
- `thickness_dec` udává hodnotu, o kterou je průměr snížen po každém průchodu. Hodnota 0, záporná nebo hodnota větší než je průměr snižuje průměr po každém průchodu na polovinu.
- `step` udává krok průchodu snímkem.
- `step_dec` udává hodnotu o kterou bude krok snížen po každém průchodu. Hodnota 0, záporná nebo hodnota větší než je krok snižuje krok po každém průchodu na polovinu.
- `pass` udává počet průchodů snímkem.
- `threshold` udává velikost prahu.
- `minL` udává minimální počet segmentů tahu.
- `maxL` udává maximální počet segmentů tahu.
- `edges` uvádí, zda budou zvýrazněny ve výstupu hrany. Možné hodnoty jsou `true` a `false`.
- `fps` udává počet snímků za sekundu ve výstupním videu.

6.4 Vývojové prostředí

Protože bylo původně uvažováno pro aplikaci grafické uživatelské rozhraní, probíhal vývoj ve vývojovém prostředí Qt Creator. Jde o velmi přívětivé prostředí usnadňující vývoj aplikace. Pro propojení s knihovnou OpenCV, stačí přidat do projektového souboru složku s include soubory a použité dynamické knihovny, viz. následující kód.

```
INCLUDEPATH += C:\\OpenCV-2.3.1\\install\\include
LIBS += -LC:\\OpenCV-2.3.1\\install\\lib \
        -lopencv_core231.dll \
        -lopencv_highgui231.dll \
        -lopencv_imgproc231.dll \
```

V aplikaci nicméně bylo upuštěno od GUI, a proto nevyužívá prvky Qt, takže k běhu ani překladu není vyžadováno mít Qt nainstalované.

Kapitola 7

Zhodnocení výstupů a výsledků

V této kapitole bude uvedena časová náročnost zpracování a zhodnoceny dosažené výsledky.

7.1 Časová náročnost

Doba zpracování videa závisí na zvolených parametrech. Obecně je největší doba zpracování při použití křivek. V případě křivek pak velice závisí na zvolené délce tahu. Čím je zadaná délka větší, tím více bodů křivky musí být spočítáno. Dobu samozřejmě ovlivňuje také počet průchodů obrázkem. Doba zpracování byla zjišťována na stolním PC s procesorem Core2Duo 3.0Ghz. Při zpracování pomocí křivek a složených přímek byl nastaven minimální počet segmentů na 2 a maximální na 18. Počet průchodů byl ve všech případech 3 a krok průchodu 12 (zmenšení o 4 po každém průchodu).

Zpracováváno bylo video o 125 snímcích, což odpovídá asi 8 sekundám při 15 snímcích za sekundu.

První tabulka (7.1) uvádí rozdíl v době zpracování při použití odlišných tahů. Délka tahu byla nastavena na 30px a průměr tahu na 12px. Je bez překvapení, že nejrychleji jsou zpracovány jednoduché přímky.

Typ	**Doba zpracování**
přímka	5
složená přímka	21
křivka	95

Tabulka 7.1: Délka 30, průměr 12, práh 25.

Při druhém měření (7.2) byla ponechána pevná délka tahu. Video je pak víceméně nepoužitelné, výsledky jsou uvedené spíše pro doplnění. Délka tahu byla nastavena na 25px. Průměr tahu byl 12px.

Typ	Doba zpracování
přímka	6
složená přímka	23
křivka	119

Tabulka 7.2: Délka 25 (neměnná v průchodech), průměr 12, práh 25.

V dalším pokusu (7.3) byla nastavena délka na 25px a průměr na 12px s tím, že se obě hodnoty zase snižují po jednotlivých průchodech na polovinu. Navíc byl přidán parametr pro řazení tahů podle jasu. Jak je vidět, procházení seznamu s pořadím tahů a jeho řazení zvyšuje podstatně dobu zpracování u přímek a složených přímek. U křivek je rozdíl zanedbatelný.

Typ	Doba zpracování
přímka	45
složená přímka	78
křivka	124

Tabulka 7.3: Délka 25, průměr 12, práh 25, řazení tahů podle jasu.

Předposlední tabulka (7.4) ukazuje vliv parametru prahu. Práh byl nastaven na hodnotu 5. Ostatní hodnoty jsou stejné jako při tabulce č. 1. Z hodnot je vidět mírný nárust u jednoduchých a složených tahů. Naopak u křivek se jedná o pokles času, potřebného ke zpracování.

Typ	Doba zpracování
přímka	8
složená přímka	23
křivka	80

Tabulka 7.4: Délka 30, průměr 12, práh 5.

Poslední tabulka (7.5) ukazuje že přidání zvýrazněných hran do výstupního videa nemá žádný vliv na čas potřebný ke zpracování (nebo je vliv zanedbatelný a projevil by se při zpracování delšího videa).

Podle časových hodnot lze říct, že při použití přímek jako tahů, lze video zpracovávat v reálném čase (8-mi sekundové video je zpracováno za 5-6 sekund). V případě složených tahů a křivek už toto možné není. Stejně tak, pokud jsou navíc při běhu programu procházeny seznamy s vygenerovanými tahy, a je prováděno jejich řazení, není možné video zpracovat v reálném čase ani při použití přímek.

Typ	Doba zpracování
přímka	6
složená přímka	22
křivka	95

Tabulka 7.5: Délka 30, průměr 12, práh 25, zvýrazněné hrany.

7.2 Grafické výstupy

V této sekci budou porovnány vždy 2 po sobě jdoucí snímky a to jak na sebe navazují. Parametrů se dá nastavit nespočet, ale ne všechny možnosti jsou vhodné pro video. Snažil jsem se proto vybrat videa zpracovaná, podle mého názoru, s vhodnými parametry a zachycující např. pohyb. Bohužel rozlišení zde není příliš vhodné pro detailní zkoumání obrázků, proto mohou některé skutečnosti pozorovateli uniknout.

Nejdříve jsem vybral pro porovnání snímky z videa, kde je kamera umístěná staticky. Pozadí je statické a v popředí se pohybuje objekt (cyklista). Video bylo převzato ze stránek skupiny „The Procrastinators“ na MIT (http://projects.csail.mit.edu/procrastinators/).

Obrázek 7.1: Originální snímky prvního videa.

Na prvním obrázku (7.2) jsou jako tahy použité křivky. Průměr je v průchodech [8,4,2], takže výsledná malba má větší detaily než obrázek 7.3. Při bližším zkoumání je možné rozpoznat drobné změny v místech, kde není mezi snímky žádný rozdíl. To je bohužel způsobeno občasným vykreslením tahu tam, kde by být neměl a v dalších snímcích se pak tato chyba může kumulovat.

Obrázek 7.2: Upravené snímky prvního videa. Křivky, délka 20, průměr 8, práh 20.

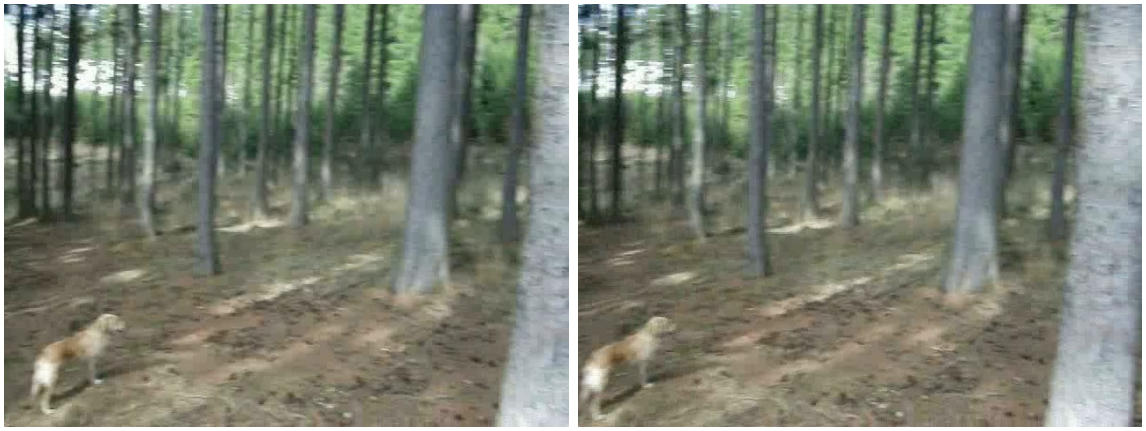
Na následujícím obrázku jsou použity složené přímky, průměr je [12,6,3]. Malba je méně detailní než při použití křivek (za což můžou i rozdílné parametry). Rušení mezi snímky je o něco menší než při použití křivek.

Obrázek 7.3: Upravené snímky prvního videa. Složené přímky, délka 16, průměr 12, práh 25.

Další obrázek (7.4) ukazuje přidané zvýrazněné hrany. Hrany dále redukují rušení způsobené pohybem tahů.

Obrázek 7.4: Upravené snímky prvního videa. Křivky, délka 22, průměr 14, práh 25, hrany.

Další snímky jsou z videa, kde se kamera pohybuje a snímá okolí (les). Ve videích, ze kterých pochází následující snímky, už byla patrnější nestabilita. Ta je způsobena pohybem kamery, protože pak se mění celý obraz.



Obrázek 7.5: Originální snímky druhého videa.

Obrázek 7.6 ukazuje snímky vymalované tahy s průměrem [8,4,2].

Obrázek 7.6: Upravené snímky druhého videa. Křivky, délka 20, průměr 8, práh 20.

Poslední obrázek (7.7) ukazuje výstup s větším průměrem štětce a o málo větším prahem. Je na něm vidět např. že země není překreslována drobnějšími tahy, protože barvy tahů se vlezou pod práh. Rušení mezi snímky, je už vcelku velké (i díky délce tahů).

Obrázek 7.7: Upravené snímky druhého videa. Křivky, délka 22, průměr 14, práh 25.

Co z obrázků není na první pohled jasně patrné, je právě koherence mezi snímky. Nejlepší výsledky dává zpracování s malou délkou a průměrem. Pak se ale snímky blíží svým rozostřeným originálům a oddalují se od nerealismu. Navíc mohou ve snímcích vznikat mřížky, které nevypadají dobře. Čím jsou pak tahy štětce delší, tím je video méně stabilní a více „bliká“.

Kapitola 8

Závěr

Cílem práce bylo nastudování vybraných technik pro nerealistické zpracování videa a následná implementace zvolené techniky. V průběhu práce byla implementována aplikace demonstrující zpracování videa nerealistickým způsobem. Důležitým bodem práce bylo zajištění základní stability videa (koherence mezi snímky), které však u použití malířských technik není triviální. Vybraná technika překreslování snímku v místech, kde se liší od předchozího, se ukázala náchylnou nejen na rychlejší změny ve snímcích vstupního videa. Při vykreslování stačilo, aby v místě, kde je nepatrná změna od předchozího snímku, vznikl nový tah, který pak působil rušivě. Tato chyba se pak často akumulovala a způsobovala překreslování snímku v místech, kde je stejný jako předchozí snímek. Řešením této situace by bylo použít jinou techniku pro zajištění stability videa, např. pomocí optického toku, nebo tyto techniky správným způsobem zkombinovat. Dobré výsledky přinášelo zpracování s nízkými hodnotami některých parametrů (nízká délka, pro tahy nepřesahující hrany objektů atd. a menší průměr štětce, kdy je obraz detailnější). Videa demonstrující zpracování je možné nalézt na CD nosiči, viz. příloha A.

Výstupy aplikace byly předvedeny několika osobám. Na zpracovaná videa byly ohlasy většinou kladné, i když si pozorovatelé všímali nestability mezi snímky. Samostatné obrázky pak měly kladný ohlas téměř vždy.

Z hlediska dalšího vývoje by mohlo být vylepšeno umísťování tahů na obrázek, které je prozatím prováděno pevně daným krokem, a tedy v mřížce, což je na některých snímcích vidět. Možností by bylo uchovávat jednotlivé tahy v paměti a nové tahy přizpůsobovat již vygenerovaným tak, aby se pokud možno nekřížily, ale jen se doplňovaly nebo lehce překrývaly. Dalším rozšířením by mohlo být přidání GUI, i když by se nejspíše jednalo o jednoduché okno s parametry, přispělo by to k pohodlnějšímu ovládání. Aplikace by se dala rozšiřovat i po stránce interaktivity, kdy by bylo možné upravovat např. směr vykreslených tahů a tím dosahovat kvalitnějšího výstupu. Zajímavým rozšířením by také bylo použití textur pro tahy namísto čar. Vylepšena by mohla být časová náročnost, kdy by mohlo být zpracování snímků přesunuto na grafickou kartu. Práce by také mohla sloužit jako základ při studiu a vývoji interaktivní aplikace pracující na principu malířských technik.

Literatura

[1] HAYS, J. a ESSA, I. Image and Video Based Painterly Animation. *GRAPHITE 2004*. 2004. S. 30–38.

[2] HERTZMANN, A. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. *SIGGRAPH 1998*. 1998. S. 453–460.

[3] HERTZMANN, A. *Algorithms for Rendering in Artistic Styles*. 2001. 159 s. Disertační práce.

[4] JOHANNESOVÁ, D. *Nerealistické zobrazení videa*. Brno: FIT VUT v Brně, 2011. 52 s. Diplomová práce.

[5] KANG, H., SEUNGYONG, L. a K. CHUI, C. Flow-Based Image Abstraction. *IEEE Transactions on Visualization and Computer Graphics*. 2009. S. 62–76.

[6] LITWINOWICZ, P. Processing Images and Video for An Impressionist Effect. *SIGGRAPH 1997*. 1997. S. 407–414.

[7] LU, J., SANDER, P. V. a FINKELSTEIN, A. Interactive Painterly Stylization of Images, Videos and 3D Animations. *Proceedings of I3D 2010*. 2010. S. 8.

[8] O'DONOVAN, P. a HERTZMANN, A. AniPaint: Interactive Painterly Animation from Video. *IEEE Transactions on Visualization and Computer Graphics*. 2012, roč. 18, č. 3. S. 475–487.

[9] REYNOLDS, C. *Stylized Depiction in Computer Graphics* [online]. Dostupné na: <http://www.red3d.com/cwr/npr/>.

[10] WANG, J., XU, Y., SHUM, H.-Y. et al. Video Tooning. *SIGGRAPH 2004*. 2004. S. 574–583.

[11] WINKENBACH, G. a SALESIN, D. H. Computer-Generated Pen-and-Ink Illustration. *SIGGRAPH 1994*. 1994. S. 11.

[12] ŽENKA, R. *Algoritmy pro nefotorealistické zobrazování*. Praha: Matematicko-fyzikální fakulta UK v Praze, 2001. 68 s. Diplomová práce.

[13] ŽÁRA, J., BENEŠ, B., SOCHOR, J. et al. *Moderní počítačová grafika*. Brno: Computer Press, 2004. 576 s. ISBN 80-251-0454-0.

Příloha A

Obsah CD

- `/src/` – Zdrojové kódy programu.
- `/bin/` – Přeložené spustitelné soubory.
- `/examples/` – Ukázková videa a obrázky.
- `/report/` – Tato práce ve formátu PDF a zdrojové texty v LaTeXu
- `README.txt` – Informace o aplikaci.